0705 컴퓨터 비전 (= 영상처리)

\*Vector Space

원점을 포함하고 있어야 됨.  
원점을 지나는 평면 위의 2D, 3D 벡터들은 어떤 수식에도 원점을 지남  
Linear combination: basis vector들에 상수배, + -  
function space: 2D, 3D 벡터들의 공간처럼 함수들도 공간을 이룸, 함수들간 + -해도 공간을 벗어 나지 않음  
-3D vector space를 표현하는데 3개의 basis vector가 있으면 3D를 표현가능  
=> basis vector is not unique  
-x+y+z=0(원점을 지나는 3D plane)  
 2개의 bases vectors로 3D plane표현 가능  
Function space  
Taylor series  
  
-Gram-Schmidt orthogonalization  
두 벡터를 내적하면 한벡터의 길이가 나오고 그 차이를 구하면 두번째 벡터가 나오고 수직으로 이뤄진 두벡터를 각각 내적해주면 나머지 한 벡터를 구한다?  
-Eigenvetors(고유 벡터)  
선형 변환이 일어난 후에도 방향이 변하지 않는, 영벡터가 아닌 벡터이다. 

1.Image Acquisition

Image: 3차원에 해당하는 신호를 2차원 공간에 투영 한 것  
이미지 신호는 함수와 같다.  
샘플링: 연속 신호(유동적인 신호)를 이산 신호(수치화된 신호)로 감소시키는 것을 말한다. 이를테면 파동 (연속 시간 신호)을 일련의 표본(이산 시간 신호)으로 바꾸는 것을 들 수 있다.  
촘촘하게 샘플링하는 것이 왜곡(신호 처리에서 표본화를 하는 가운데 각기 다른 신호를 구별해내지 못하게 하는 효과)을 줄여줌

noisy의 문제는 눈에 보이지 않지만, 존재함 => 생각한 결과가 나오지 않는 경우가 생김  
nosiy를 어떠한 식으로 조금이라도 제거해서 학습할지에 대해 고민할 것  
초음파 사진: 노이즈가 굉장히 심하고, 해상도가 낮다

이미지는 배열구조로 구성됨  
녹색이 더 높은 비율로 샘플링이 된다.

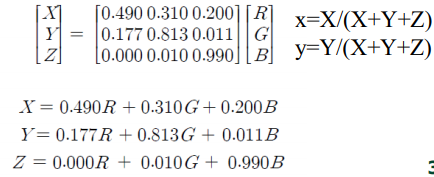
이미지 해상도: 픽셀이 클수록 해상도가 증가, 노이즈를 감소시켰을 때 증가,   
양자화: 샘플링한 아날로그 형태로 되어 있는 신호나 정보를 디지털화 하는 작업

-Functional forms

f(x,y): 2D image  
f(x,y,t), f(x,y,z): video sequence, 3D model

ν Spatial sampling by pixel grid :  
ν Quantization of intensities :  
ν Noise reduction :  
ν Demosaic :  
ν Contrast enhancement :  
ν Color enhancement :  
ν Radiometric compensation :  
ν White balance, anti-vignette :  
ν Lens distortion compensation :  
ν Compression:

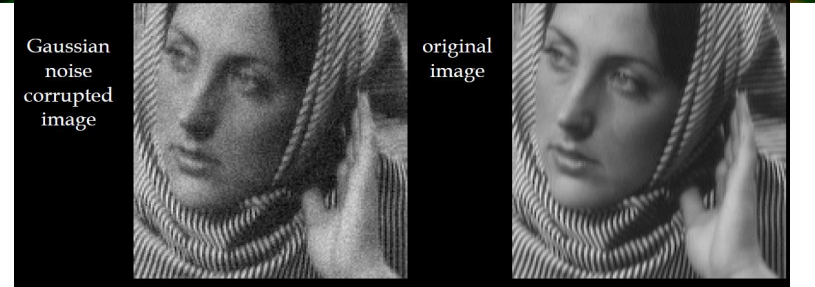
2.color space image measure

Additive matching: M = aP1+bP2+cP3   
M이 color, P1,P2,P3: RGB, 함수 basis, 스펙트럼  


Y: 밝기, UV: 색 컴포넌트, RBG와 역행렬, U:Cb(b와의 차이), V:Cr(r과의 차이), 밝기에 신경을 써야할 때 사용됨

PSNR이 커지면 커질수록 원본 영상과 유사해 진다.

SSIM이 MSE보다 더 중요하다.

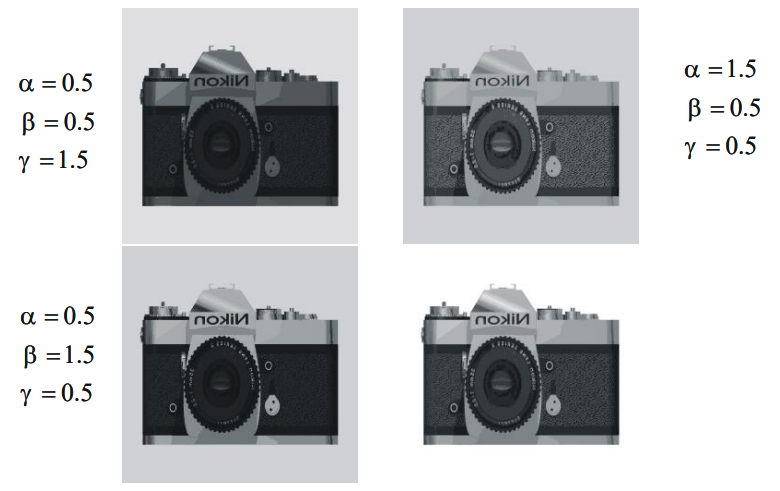
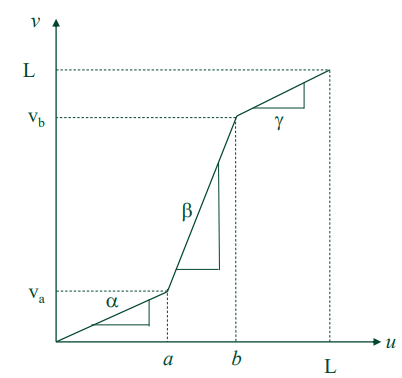


줄무늬는 SSIM이 높고 뺨은 낮음

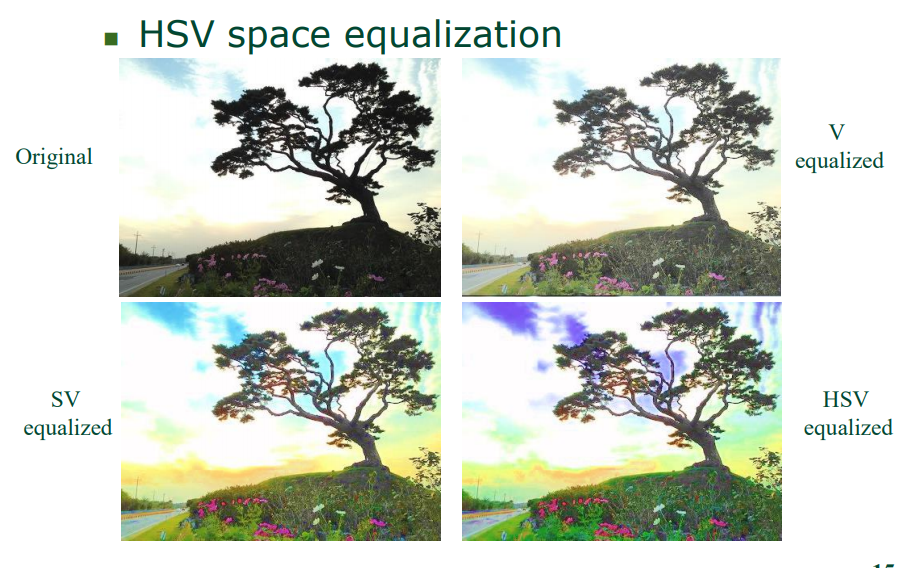
3.Image Enhancement

Main topics

ν Contrast enhancement   
ν Noise reduction  
ν Sharpening   
ν Colorization

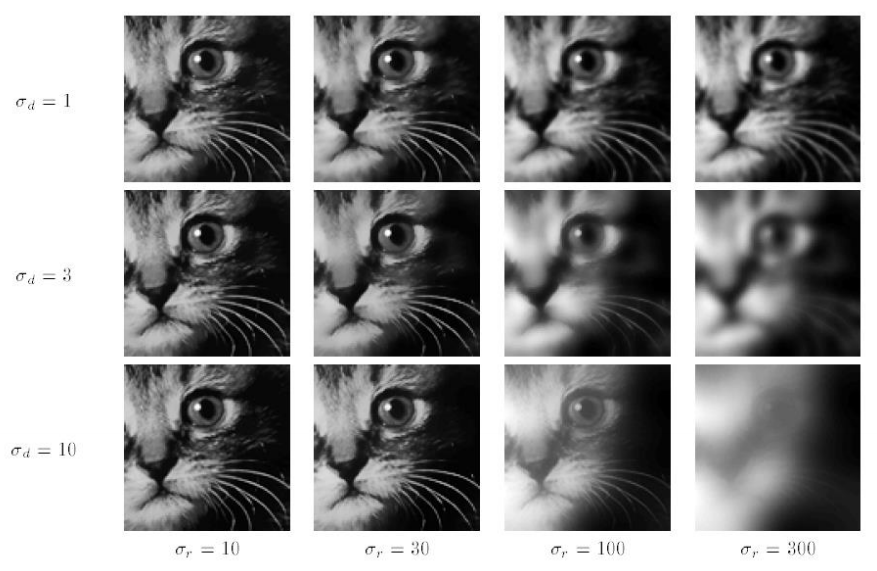
명도 대비  


\*과제



-Bilateral Filter (Denoise Filter)

영상의 흐릿함을 제거하기 위해

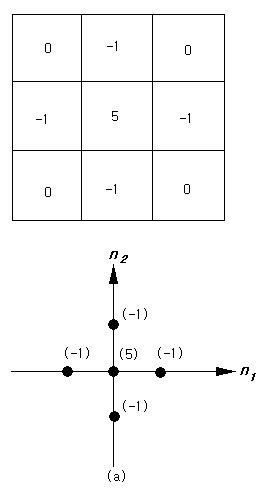
Bilateral filter => Domain + Range  
Domain: 정의역, Range: 치역(색, 가중치)  


- Median Filtering



특정 범위에서 가장 많은 값들이 있는 색으로 채움

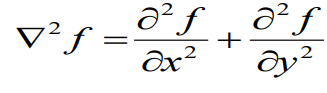
-High Pass Filtering



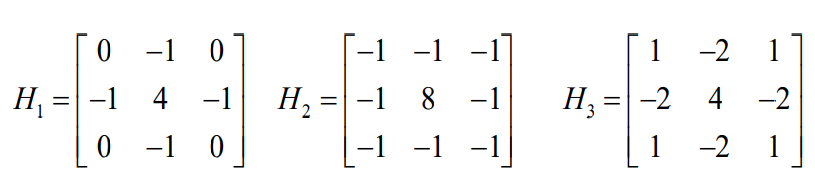
총합은 1

-Unsharp Mask





기울기가 급격하게 변하는 점



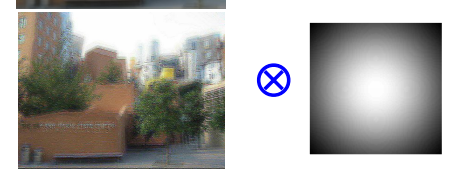
계수들의 총합은 0



어두운 부분은 주변보다 어둡게, 밝은 부분은 더 밝게 해서 선명하게 보이도록 함  
0706

4.Image Restoration

blurring: 초점이 맞지 않아 흐릿하게 보이는 경우, 심하면 심할수록 blur가 제거된 이미지는 얻기 힘들어 짐

-가우시안 커널  


(1)Image Deblurring  
(2)Image Deconvolution  
(3)Methodology  
Concolution: 입력을 어떤 특성을 갖는 시스템에 통과시켰을 때 출력을 구하는 과정

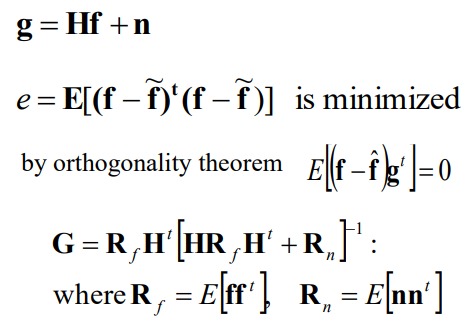
-Degradation Functions  
Blurring + noise  
Point spread function ( blurring )  
n번 진동하는 녀석의 복소수 형태의 크기

푸리에 변환(Fourier transform, FT)은 시간이나 공간에 대한 함수를 시간 또는 공간 주파수 성분으로 분해하는 변환

-inverse filter

convolution(어제 한 필터들) -> 주파수 함수 -> 역수 -> Inverse Fourier Transform -> Deblurring

-Wiener filtering: 확률신호처리에서 랜덤한 값을 결정해야되는데 무엇이 좋은 것인지 어케 결정하냐 ->

H: 커널, n: noisy  


(열벡터 형태): 차이가 작으면 작을수록 현명한 영상이 만들어 짐

백색 잡음: 모든 주파수 수준에서 동등한 잡음을 가지는 것

-Constrained Least Squares

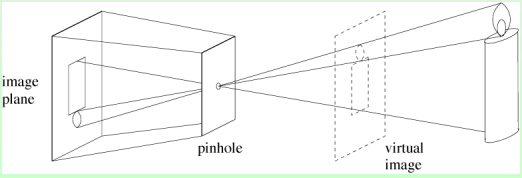
(ax1 + by1 = c1)^2  
(ax2 + by2 = c2)^2  
(ax3 + by3 = c3)^2 더한 값 = | Ax -b |^2

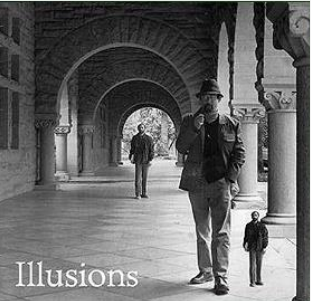
실제 데이터와 직선 혹은 곡선의 차를 제곱하여 더한 값이 최소가 되도록 하는 것

0707 cv

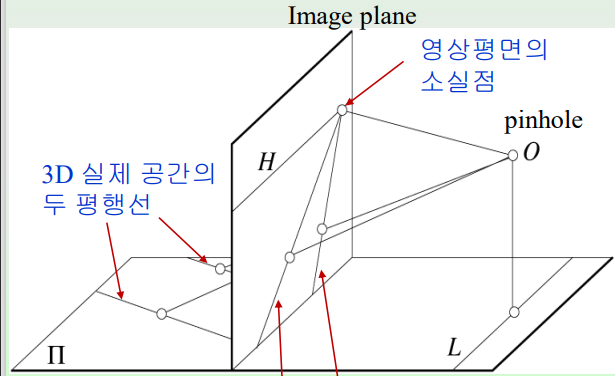
6.Geometric Camera Models

-Pinhole Camera



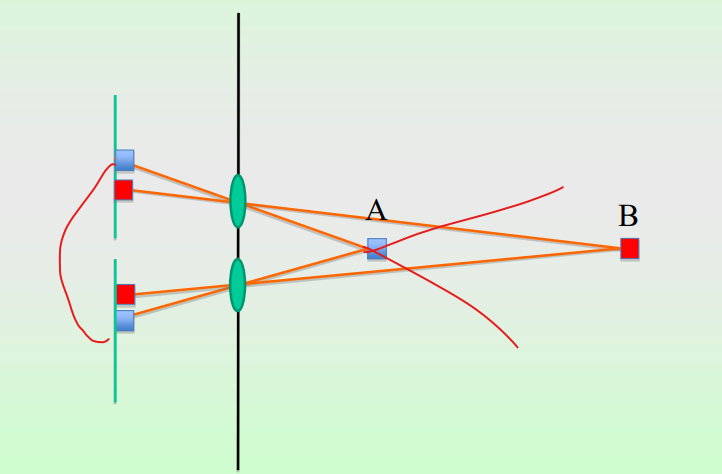
상하좌우 반대로 상이 맺힘  
거리에 따라 투영되는 위치나 크기가 변함  


공간정보를 없애버리면 Perspective Effects



공간의 평행한 모든 직선은 한 점으로 모임 => 소실점  
평행하게 비스듬히 봄으로써 소실점이 빠르게 형성될 수 있음  
수평선: 바라보는 시선의 높이

- Two Pinholes(스테레오 카메라)



가까이 있는 물체는 픽셀차이가 크고, 멀리 있으면 작음

삼각측량?: 물체와의 거리를 계산

렌즈 사용이유: 많은 양의 빛을 빠르게 모아 상을 맺게 해 줌  
빛이 한곳에 안 맺히면 blurring 발생 but 지금까지 잘 해결해서 렌즈 사용 중

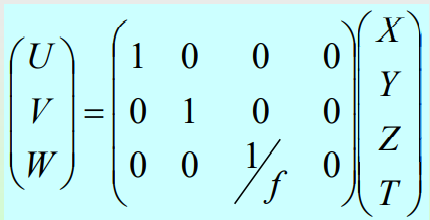
f(x+y) = f(x) + f(y)  
f(kx) = k f(x) => Homogeneous  
linear 조건

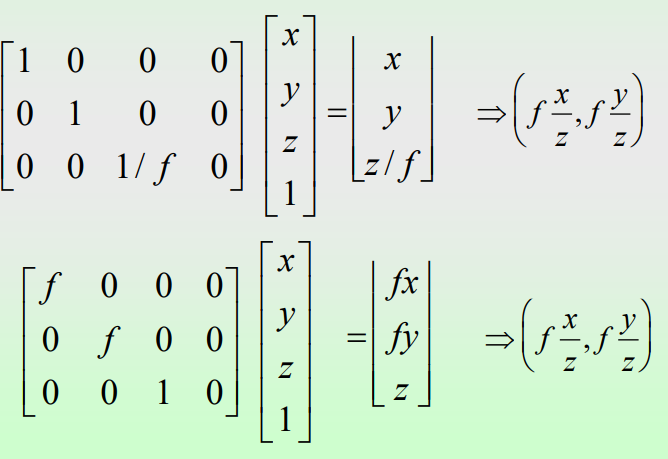
-Homogeneous Coordinates

2차원의 경우에는 (x, y)를 (x, y, 1)나 (wx, wy, w)로 표현  
3차원의 경우에는 (X, Y, Z)를 (X, Y, Z, 1) 나 (wX, wY, wZ, w)로 표현

3D => 2D  
(x, y, w)를 (x/w, y/w)  
w가 0일 때 그 점은 무한대에 있고 소실점

Degrees of freedom (DOF): 독립적인 미지수의 개수가 몇 개인가

-Equation of Projection  


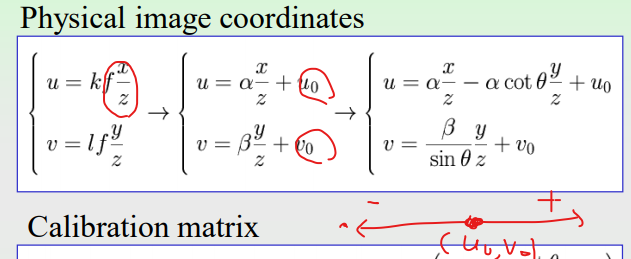


어떻게 하든 Homogeneous에 의해서 한점이 구해짐

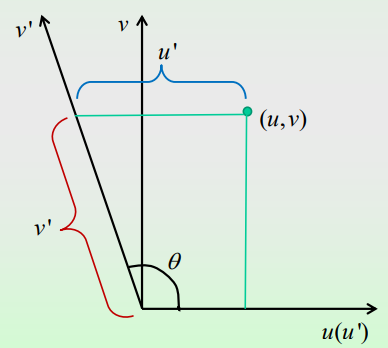
- Camera parameters

focal length f:   
principal point (x’ c , y’ c ):  
pixel density (sx , sy ): 센서의 규격  
skew angle (θ): 90도

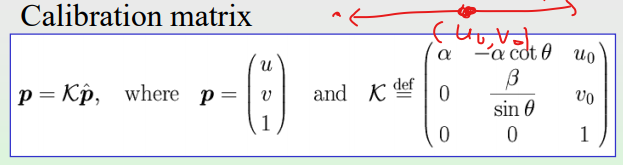
렌즈 자체의 왜곡은 렌즈의 크기나, 볼록렌즈의 초점거리나 근접하게 찍으려고 할 때 생김



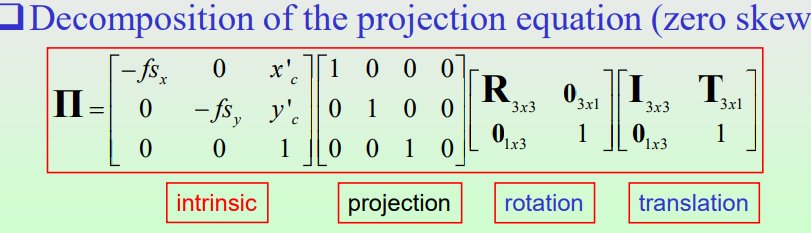
u0, v0를 픽셀 중심 좌표로 하고 u, v에 더해주면 픽셀좌표 표현됨



θ가 90이상으로 늘어났을 때, 기존 u라고 생각한 위치에 θ-90만큼(u + u tan(θ- π/2) 더해줌



픽셀좌표로 변환해 주는 행렬(θ를 90으로)

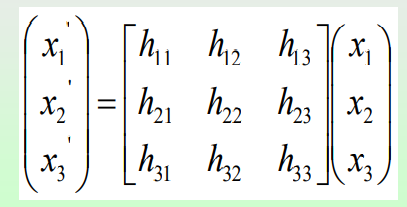


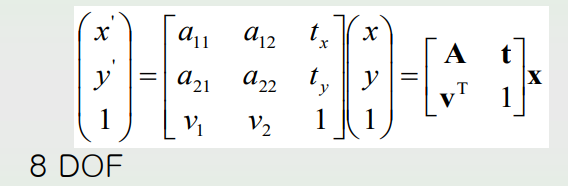
3차원 좌표를 2차원 픽셀좌표로 투영시키는 행렬식

-2D Projective Transformation=> homography

평평한 평면, 영상을 3D에서 보는 것처럼 변환하고 싶을 때 사용







v1, v2가

- Radial Distortion

볼록렌즈의 굴절률에 의한 것으로서 왜곡 정도가 중심에서의 거리에 의해 결정되는 왜곡

-Camera Calibration

3차원 점들이 영상에 투영된 위치를 구하거나 역으로 영상좌표로부터 3차원 공간좌표를 복원할 때에는 이러한 내부 요인(사용된 렌즈, 렌즈와 이미지 센서와의 거리, 렌즈와 이미지 센서가 이루는 각 등 카메라 내부의 기구적인 부분)을 제거해야만 정확한 계산이 가능해집니다. 그리고 이러한 내부 요인의 파라미터 값을 구하는 과정을 **카메라 캘리브레이션**

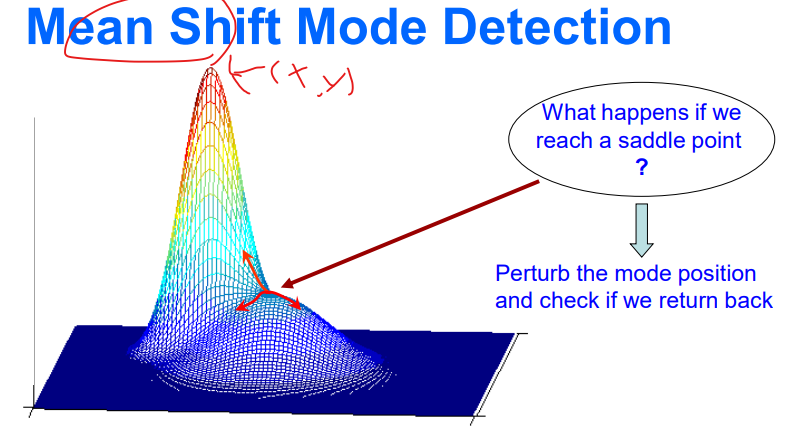
**-** Mean Shift

처음 중심점을 잡고 밀도가 높은 쪽으로 계속 이동시킴.  
위 방법을 반복하며 가장 밀도가 높은 지점 하나를 택함  
핵심: 확률적으로 가장 발생가능성이 높은 벡터를 찾음

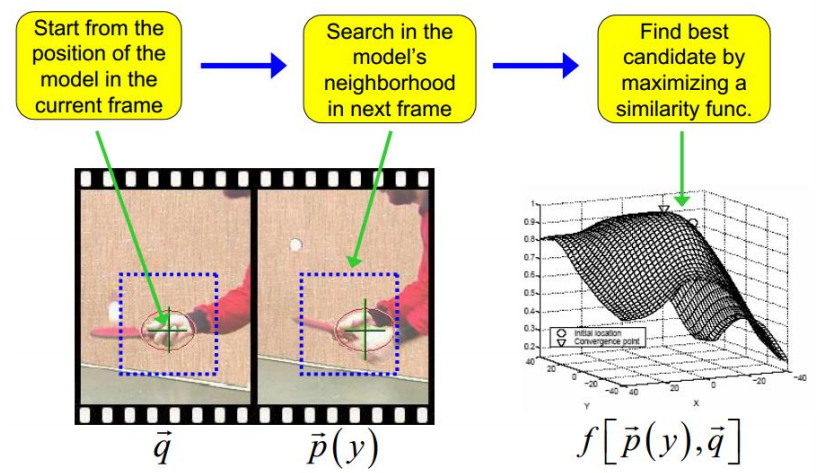
K-mean, mean shift 모두 클러스터링

Kernel Density Estimation Kernel Density Estimation Gradient

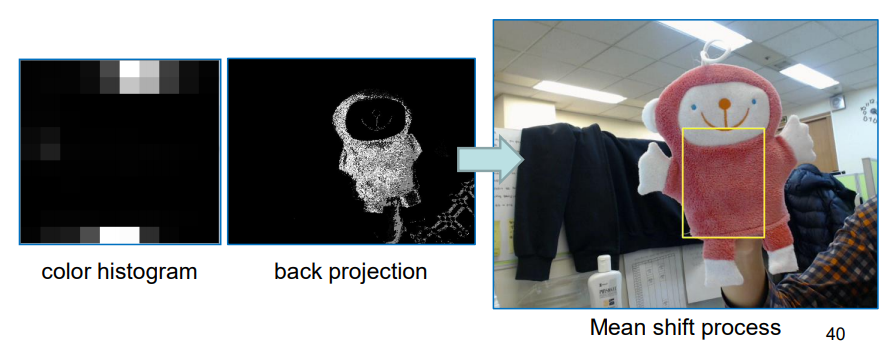
- Mean Shift



sad point에서는 일시적으로 기울기가 0에 가깝기에 오해할 수도 있음  
한번 수렴했다 하더라도 다른 initial point를 잡고 다시 mean shift를 돌렸을 때 다시 그 자리로 오면 그 점은 mean shift, 그렇지 않으면 local min에 빠져 있는 것



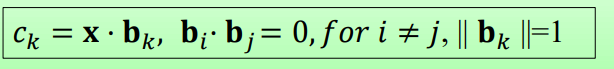
트랙킹은 위치좌표를 찾는 것이 핵심, 위치과 크기에 초점을 맞출 것  
오른쪽은 물체의 유사한 정도를 나타낸 것  
젤 위는 가장 유사한 위치의 좌표

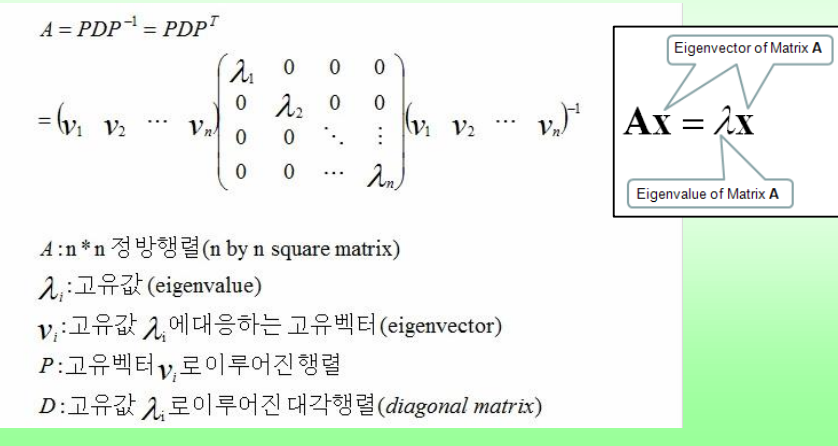


back projection: tracking하고자 하는 이미지에 대해서 해당되는 컬러 히스토그램 q에 대하여 각각의 컬러를 assign 하는 것, 분홍색과 가깝지 않은 색상은 0으로 assign해서 안 보임

opencv를 통해 tracking을 해보면 색상의 확률 분포가 균일할 시 잘 안됨, color만으로 tracking을 하니 문제가 생김 => 물체의 크기를 어느정도 기준으로 잡고 tracking을 하는 방법을 추가해서 구현

-Basis Vectors



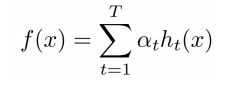


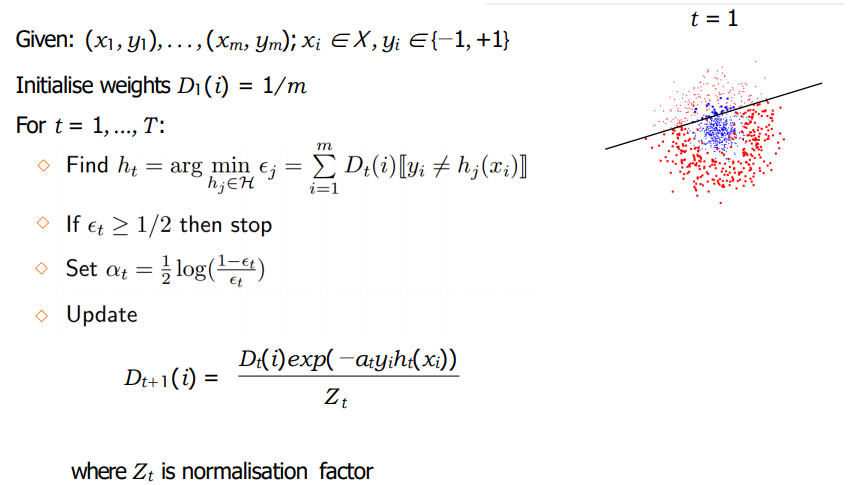
**그람-슈미트(Gram-Schmidt) 직교화란?**

주어진 벡터들을 이용해서 서로 수직인 벡터들을 만드는 방법이다. 좀더 고상한 말로 표현하면 주어진 벡터들에 대한 직교기저(orthogonal basis) 또는 정규직교기저(orthonormal basis)를 구하는 과정이다.

16.boosting

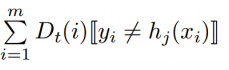
딥러닝이 뜨기 전 가장 많은 사람들이 활용하려고 했던 학습방법  
AdaBoost: 적응적으로 부스팅  
최적으로 최고의 성능을 발휘할 수 있도록 조합으로 함으로써 strong을 만든다  
0을 기준으로 ~한 연산을 했을 때 0보다 작으면 -, 크면 +

  
x: 인식하고자 하는 데이터



m개의 2차원 좌표가 주어지고 class를 판별하고   
처음이니까 1/m이라는 가중치를 줌  
무수히 많은 직선들을 다 뽑아보면서 가장 파란색과 빨간색을 제일 잘 구분해주는 하나의 직선을 찾음  
  
class를 정하는 알고리즘  
ht(x) => 1 or -1  
at: 항상 양수  
yi \* ht(xI) => 올바르게 인식하면 1, 인식 실패 시 -1  


최종적으로 인식에 실패하면 양수(weight를 키워줌), 성공하면 음수(weight를 줄여줌)



인식에 실패한 weight의 합을 error를 정의하고 그 error가 최소가 되는 직선을 찾음

가중치를 추가하는 것이 핵심 그래서 adaptive라고 함

특정한 ht가 인식률이 높으면 error가 작고 가중치가 큼

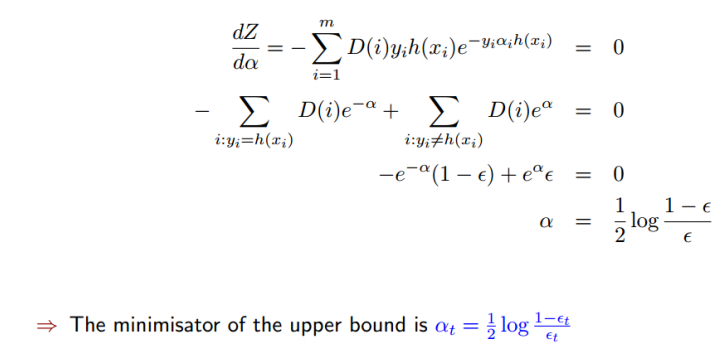
가중치가 하나하나 추가되면서 성능을 향상 => boosting

무조건 weight가 커졌다고 해서 문제가 다 해결되는 것은 아님

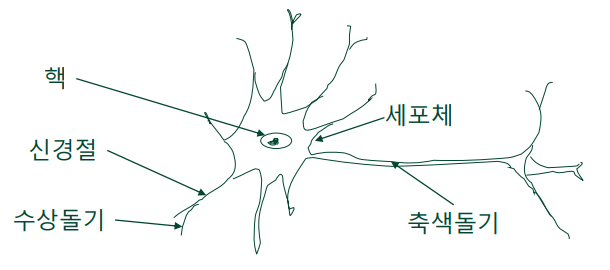
-Upper Bound Theorem

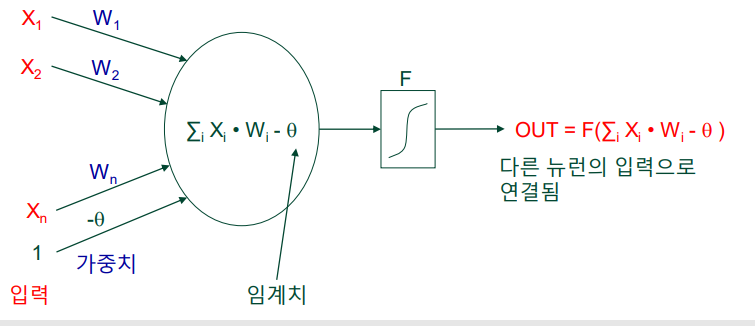
총 m개를 만든 strong classify를 m으로 나눠서 성공률?을 구함  
실패한 에러를 최소화 시키려면 에러를 직접 컨트롤하는 것이 아니라 upper bound가 최소한이 되게 하여 에러를 최소화 시키겠다. => Zt를 최소화 시키는 방향으로

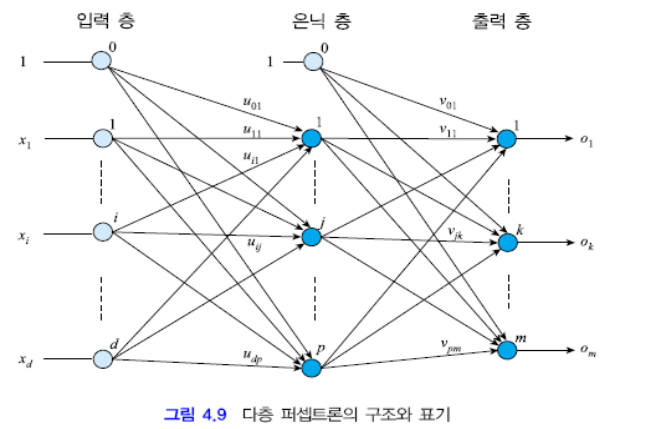


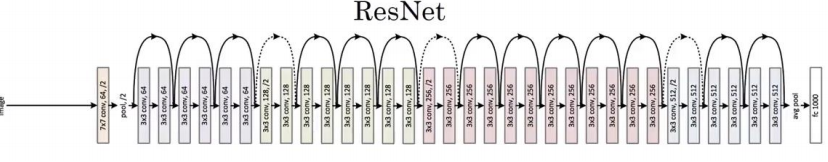


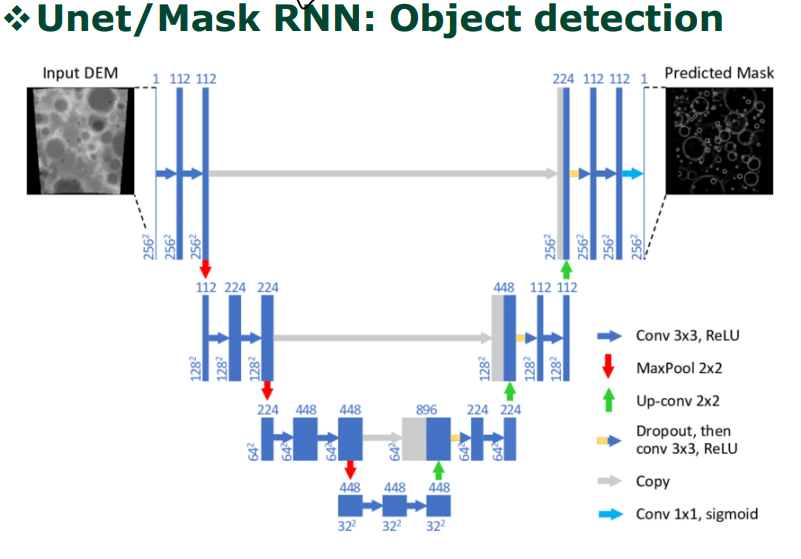
인식의 성공 유무에 따라 식을 나눔.



수상돌기: input 신호  
세포체에서 신호를 처리  
축색돌기: output  




대부분의 연산은 은닉 층에서, 학습한다 => 가중치와 입력을 연산해서 합산한다.  
DNN: Hidden layer의 수를 많게 함  
CNN: 특정한 필터들을 노드의 입력란에 출력 결과값으로 넣어주는 것, 이미지에 해당하는 처리를 할 때 많이 쓰임  
  
layer를 건너뛰면서 해상도가 낮아지면서 찾지 못하는 문제를 해결



비정형적인 영역을 표시하고 학습해서 그 영역을 좀 더 섬세하게 분할해줌.

사람의 형상을 박스로 구분하는 것이 아니라 형상을 그대로 따서 구분 가능

전지전능한 네트워크 모델은 없다.  
목적과 시스템에 맞는 네트워크 선정 및 설계

동일한 네트워크를 이용하더라도 학습 데이터와 학습 방법에 따라 결과가 크게 달라질 수 있다.  
다양한 조합의 학습 데이터  
학습 대상의 크기/해상도에 따른 차이

학습데이터가 많을수록 좋은 것은 아니다.  
학습에 일관성을 부여하는 데이터가 중요  
예외적인 학습 범위를 넘는 오류는 선별